

2651

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re U.S. Patent Application of
HAMASAKI et al

Application Number: 09/977,334

Filed: October 16, 2001

**For: DIGITAL DATA DECOMPRESSING
SYSTEM AND METHOD**



RECEIVED
DEC 10 2001
Technology Center 2600

Honorable Assistant Commissioner
for Patents
Washington, D.C. 20231

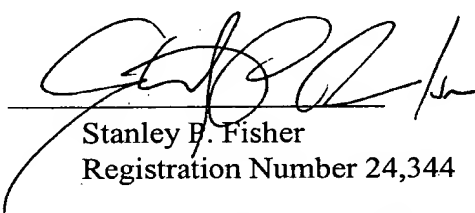
LETTER

Sir:

The below-identified communications are submitted in the above-captioned application or proceeding:

<input checked="" type="checkbox"/> (X)	Priority Documents (1)	<input type="checkbox"/> ()	Verified English Translation
<input checked="" type="checkbox"/> (X)	Notice of Priority	<input type="checkbox"/> ()	Information Disclosure Statement
<input type="checkbox"/> ()	Response to Missing Parts with executed declaration	<input type="checkbox"/> ()	Notice of Related Applications
		<input type="checkbox"/> ()	Check for \$.00

The Commissioner is hereby authorized to charge payment of any fees associated with this communication, including fees under 37 C.F.R. § 1.16 and 1.17 or credit any overpayment to **Deposit Account Number 08-1480**. A duplicate copy of this sheet is attached.


Stanley B. Fisher
Registration Number 24,344

REED SMITH HAZEL & THOMAS LLP
3110 Fairview Park Drive
Suite 1400
Falls Church, Virginia 22042
(703) 641-4200

JUAN CARLOS A. MARQUEZ
Registration No. 34,072

December 6, 2001

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

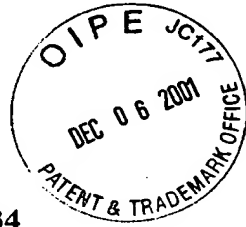
In re U.S. Patent Application of

HAMASAKI et al

Application Number: 09/977,334

Filed: October 16, 2001

For: **DIGITAL DATA DECOMPRESSING
SYSTEM AND METHOD**



RECEIVED
DEC 10 2001
Technology Center 2600

Honorable Assistant Commissioner
for Patents
Washington, D.C. 20231

**REQUEST FOR PRIORITY
UNDER 35 U.S.C. § 119
AND THE INTERNATIONAL CONVENTION**

Sir:

In the matter of the above-captioned application for a United States patent, notice is hereby given that the Applicant claims the priority date of October 17, 2000, the filing date of the corresponding Japanese patent application 2000-316093.

The certified copy of corresponding Japanese patent application 2000-316093 is being submitted herewith. Acknowledgment of receipt of the certified copy is respectfully requested in due course.

Respectfully submitted,

Stanley P. Fisher

Registration Number 24,344

REED SMITH HAZEL & THOMAS LLP

3110 Fairview Park Drive

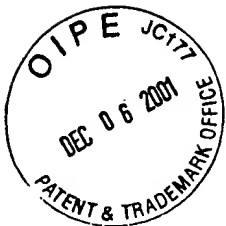
Suite 1400

Falls Church, Virginia 22042

(703) 641-4200

December 6, 2001

JUAN CARLOS A. MARQUEZ
Registration No. 34,072



日本国特許庁
JAPAN PATENT OFFICE

別紙添付の書類に記載されている事項は下記の出願書類に記載されて
いる事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed
with this Office

出願年月日
Date of Application:

2000年10月17日

出願番号
Application Number:

特願2000-316093

出願人
Applicant(s):

株式会社日立製作所

CERTIFIED COPY OF
PRIORITY DOCUMENT

2001年10月19日

特許庁長官
Commissioner,
Japan Patent Office

及川耕造

出証番号 出証特2001-3091552

【書類名】 特許願

【整理番号】 H00014561

【あて先】 特許庁長官殿

【国際特許分類】 G06F 12/04

【発明者】

【住所又は居所】 東京都小平市上水本町五丁目20番1号 株式会社日立製作所 半導体グループ内

【氏名】 浜崎 博幸

【発明者】

【住所又は居所】 東京都小平市上水本町五丁目20番1号 株式会社日立製作所 半導体グループ内

【氏名】 宮本 崇

【発明者】

【住所又は居所】 東京都小平市上水本町五丁目20番1号 株式会社日立製作所 半導体グループ内

【氏名】 武田 博

【発明者】

【住所又は居所】 東京都小平市上水本町五丁目20番1号 株式会社日立製作所 半導体グループ内

【氏名】 佐藤 潤

【発明者】

【住所又は居所】 東京都小平市上水本町五丁目20番1号 株式会社日立製作所 半導体グループ内

【氏名】 大村 賢一郎

【発明者】

【住所又は居所】 東京都国分寺市東恋ヶ窪一丁目280番地 株式会社日立製作所 中央研究所内

【氏名】 鮎川 一重

【特許出願人】

【識別番号】 000005108

【氏名又は名称】 株式会社 日立製作所

【代理人】

【識別番号】 100085811

【弁理士】

【氏名又は名称】 大日方 富雄

【電話番号】 03-3269-1430

【手数料の表示】

【予納台帳番号】 027177

【納付金額】 21,000円

【提出物件の目録】

【物件名】 明細書 1

【物件名】 図面 1

【物件名】 要約書 1

【プルーフの要否】 要

【書類名】 明細書

【発明の名称】 デジタルデータの伸張システムおよび伸張方法

【特許請求の範囲】

【請求項 1】 圧縮されたデジタルデータを伸張処理して元のデータを復元するデジタルデータの伸張処理システムであって、

上記伸張されたデータが格納される複数のメモリ領域と、該複数のメモリ領域のそれぞれに対応して設けられ格納データがすべて所定の論理値か否かを示すフラグとを備え、上記メモリ領域に書き込まれるべきデータがすべて所定の論理値のときは、対応する上記フラグを第 1 の状態に設定するようにされていることを特徴とするデジタルデータの伸張システム。

【請求項 2】 上記所定の論理値は論理“0”であることを特徴とする請求項 1 に記載のデジタルデータの伸張システム。

【請求項 3】 上記メモリ領域に格納されたデータ同士の演算を行なう演算回路を備え、上記メモリ領域に格納された 2 つのデータを加算する処理を行なう際に、上記加算される 2 つのデータのうち一方に対応する上記フラグが第 1 の状態にされているときは、フラグが第 1 の状態にされていない他方のメモリ領域のデータを読み出し演算後のデータが格納されるべき第 3 のメモリ領域に格納するようにされていることを特徴とする請求項 2 に記載のデジタルデータの伸張システム。

【請求項 4】 上記メモリ領域に格納されたデータ同士の演算を行なう演算回路を備え、上記メモリ領域に格納された 2 つのデータを乗算する処理を行なう際に、上記乗算される 2 つのデータのうち一方に対応する上記フラグが第 1 の状態にされているときは、演算後のデータが格納されるべきメモリ領域に対応する上記フラグを第 1 の状態に設定するようにされていることを特徴とする請求項 2 または 3 に記載のデジタルデータの伸張システム。

【請求項 5】 上記メモリ領域に格納されたデータ同士の演算を行なう演算回路と、上記データが格納される複数のメモリ領域を識別するための情報を管理するメモリ管理手段とを備え、上記メモリ領域に格納された 2 つのデータを加算する処理を行なう際に、上記加算される 2 つのデータのうち一方に対応する上記

フラグが第1の状態にされているときは、上記フラグが第2の状態のメモリ領域の識別情報と演算後のデータが格納されるメモリ領域の識別情報とを入れ替え、該入替えにより演算後のデータが格納されたメモリ領域に対応する上記フラグを第2の状態に設定するようにされていることを特徴とする請求項2に記載のデジタルデータの伸張システム。

【請求項6】 上記フラグが第1の状態にされている上記メモリ領域からデータを読み出す際に、当該メモリ領域のデータに代えて論理“0”のデータを出力する固定データ出力手段を有することを特徴とする請求項2、3、4または5に記載のデジタルデータの伸張システム。

【請求項7】 上記伸張されるデジタルデータは、オーディオデータであることを特徴とする請求項2、3、4、5または6に記載のデジタルデータの伸張システム。

【請求項8】 上記伸張されるデジタルデータは、画像データであることを特徴とする請求項2、3、4、5または6に記載のデジタルデータの伸張システム。

【請求項9】 伸張されたデータが格納される複数のメモリ領域と、該複数のメモリ領域のそれぞれに対応して設けられ格納データがすべて論理“0”か否かを示すフラグと、上記メモリ領域に格納されたデータ同士の演算を行なう演算回路とを備え、所定の方式に従って圧縮されたデジタルデータが所定のフォーマットのビットストリームとして入力され、該ビットストリームデータを伸張処理してデータを復元するデジタルデータの伸張処理システムにおいて、

上記メモリ領域に格納された2つのデータを加算する処理を行なう際に、上記加算される2つのデータのうち一方に対応する上記フラグが第1の状態にされているときは、フラグが第1の状態にされていない他方のメモリ領域のデータを読み出し演算後のデータが格納されるべき第3のメモリ領域に格納し、当該メモリ領域に対応する上記フラグを第2の状態に設定することを特徴とするデジタルデータの伸張方法。

【請求項10】 伸張されたデータが格納される複数のメモリ領域と、該複数のメモリ領域のそれぞれに対応して設けられ格納データがすべて論理“0”か

否かを示すフラグと、上記メモリ領域に格納されたデータ同士の演算を行なう演算回路とを備え、所定の方式に従って圧縮されたデジタルデータが所定のフォーマットのビットストリームとして入力され、該ビットストリームデータを伸張処理してデータを復元するデジタルデータの伸張処理システムにおいて、

上記メモリ領域に格納された2つのデータを乗算する処理を行なう際に、上記乗算される2つのデータのうち一方に対応する上記フラグが第1の状態にされているときは、乗算後のデータが格納されるべき第3のメモリ領域に対応する上記フラグを第1の状態に設定することを特徴とするデジタルデータの伸張方法。

【請求項11】 伸張されたデータが格納される複数のメモリ領域と、該複数のメモリ領域のそれぞれに対応して設けられ格納データがすべて論理“0”か否かを示すフラグとを備え、所定の方式に従って圧縮されたデータが所定のフォーマットのビットストリームとして入力され、該ビットストリームデータを伸張処理してデータを復元するデジタルデータの伸張処理システムにおいて、

有効なデータ数を計数しながら伸張された有効データを上記複数のメモリ領域に順次格納するとともに該メモリ領域に対応する上記フラグを第1の状態にさせ、有効データがなくなった場合に残りのメモリ領域のうち復号処理されるデータが格納されるべきメモリ領域に対応する上記フラグを第2の状態に設定することを特徴とするデジタルデータの伸張方法。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

本発明は、圧縮データの伸張技術に関し、例えばMPEG1オーディオ規格に従って圧縮されたオーディオデータを伸張するオーディオデータ再生システムやJPEG規格に従って圧縮された画像データを伸張する画像データ再生システムなどに利用して有効な技術に関する。

【0002】

【従来の技術】

MPEG1オーディオ規格は、国際標準化機関ISOで規格化された国際規格であり、オーディオデータの圧縮・伸張に関する技術である。MPEGオーディ

オ規格に従って圧縮・伸張されたデータは、音質の劣化が少なく、そのデータ量が大きく削減されるため、MP3プレーヤーのような携帯用オーディオ再生装置などに利用されるようになって来ている。

【0003】

以下、MPEG1のレイヤIIIを例にとって、図17を用いて、圧縮されたオーディオデータを伸張する復号処理の手順について説明する。

【0004】

圧縮されたオーディオデータは、ビットストリームと呼ばれるシリアルデータとして、伸張を行なうシステムに供給される。ビットストリームは、所定のフォーマットのフレームと呼ばれるデータが時系列的に並んだものであり、各フレームにはヘッダやフォーマット情報、エラーチェックコード、ビット割り当て情報、スケールファクタ情報、サブバンドサンプルデータなどが含まれる。

【0005】

サブバンドサンプルデータは、入力信号がある時間（8m秒）毎のブロックに区切られ、32の周波数帯に分割されて変形離散コサイン変換（MDCT）、非線形量子化、ハフマン符号化がなされて圧縮されている。1ブロック分のサブバンドサンプルデータの圧縮データにヘッダ、フォーマット情報、エラーチェックコード、ビット割り当て情報、スケールファクタ情報を付加し所定の順序に並べたものが1フレームで、このフレームが時系列的に連続したものがビットストリームである。

【0006】

伸張処理では、図17に示されているように、先ずビットストリームBSからサンプルデータの部分を取り出して、圧縮処理でハフマン符号のような可変長符号に符号化されたデータを復号する復号化（ステップS1）を行ない、32の周波数帯毎のサンプルデータSPLS1～PL32を得る。この復号化されたサンプルデータは、それぞれ逆量子化テーブルを用いて量子化される前のデータに戻す逆量子化処理が行なわれる（ステップS2）。その後、周波数／時間変換処理（ステップS3）が行なわれてから、合成フィルタにより全ての周波数帯のデータの合成（ステップS4）が行なわれてPCM音源データPSDとして出力され

る。

【0007】

【発明が解決しようとする課題】

MPEG1オーディオ規格に従ってエンコード（圧縮）されたオーディオデータをデコード（伸張）する処理においては、復号化された周波数帯のデータに偏りがあり特定の周波数帯（低周波数側）にのみデータが存在し、すべてのビットが論理“0”である周波数帯のデータが比較的多く発生する。ところで、オーディオデータの再生システムでは、受信したビットストリームからデータをデコード（伸張）処理する際に、データを一旦RAMに格納してから読み出して種々の演算を行ない、演算結果を再びRAMに格納する処理が繰り返えされる。

【0008】

しかしながら、従来の圧縮されたオーディオデータの再生システムにおいては、伸張処理の際にすべてのビットが論理“0”である周波数帯のデータに関しても、RAMへのデータの書込み、読出し処理が行なわれていた。本発明者らは、このような処理方式では、RAMへのアクセスに伴う消費電力が必要以上に多くなっていることを見出した。特に、MPEG1オーディオ規格は、MP3プレーヤーのような電池で動作する携帯用電子機器で採用されることが多いので、消費電力の低減が強く望まれる。

【0009】

この発明の目的は、MPEG規格あるいはJPEG規格に従って圧縮されたデータを伸張するシステムにおいて、RAMへのアクセス回数を減らして、システムの消費電力を低減することができる圧縮データの伸張技術を提供することにある。

【0010】

この発明の他の目的は、MPEG規格あるいはJPEG規格に従って圧縮されたデータを伸張するシステムにおいて、復号されたデータに対する演算処理を簡略化して、システムの負担を軽減するとともに演算処理速度を高めることができる圧縮データの伸張技術を提供することにある。

【0011】

この発明の前記ならびにそのほかの目的と新規な特徴については、本明細書の記述および添付図面から明らかになるであろう。

【0012】

【課題を解決するための手段】

本願において開示される発明のうち代表的なものの概要を説明すれば、下記のとおりである。

【0013】

すなわち、圧縮されたデジタルデータを伸張処理して元のデータを復元するデジタルデータの伸張処理システムにおいて、上記伸張されたデータが格納される複数のメモリ領域と、複数のメモリ領域のそれぞれに対応して設けられ格納データがすべて所定の論理値（例えば“0”）か否かを示すフラグとを備え、上記メモリ領域に書き込まれるべきデータがすべて所定の論理値のときは、対応する上記フラグを第1の状態に設定するようにした。

【0014】

より具体的には、例えばMPEG規格あるいはJPEG規格に従って圧縮されたデータを伸張するシステムにおいて、圧縮されたデータを伸張する際に演算される値を格納するバッファメモリを複数のバンクに分けて、各バンクごとにバンク内のすべてのデータが論理“0”か否かを示すオールゼロフラグを設け、バンク内に書き込むべきデータがすべて論理“0”（以下、単に“0”と記す）であるときはバッファメモリへの実際の書込みを行わずにオールゼロフラグをセットするとともに、データ読出し時にはこのフラグを見てバンク内のデータがすべて“0”のときはバッファメモリからの読出しを省略するようにしたものである。

【0015】

上記した手段によれば、メモリ領域としてのバンクに格納しようとするデータがすべて“0”あるいはバンクから読み出そうとするデータがすべて“0”のときはバッファメモリに対するアクセスを行なわないので、メモリのアクセス回数が少なくなって消費電力が低減される。しかも、バッファメモリへの書込みや読出しを行なわなくても、オールゼロフラグによりデータがすべて“0”であるこ

とを記憶し、伸張の際にそのことを知ることができる。

【0016】

また、上記メモリ領域に格納されたデータ同士の演算を行なう演算回路を備え、上記メモリ領域に格納された2つのデータを加算する処理を行なう際に、上記加算される2つのデータのうち一方に対応する上記フラグが第1の状態にされているときは、フラグが第1の状態にされていない他方のメモリ領域のデータを読み出し演算後のデータが格納されるべき第3のメモリ領域に格納するようにする。

【0017】

より具体的には、あるバンクのデータと他のバンクのデータとを加算しようとするときに、オールゼロフラグを見て一方のバンクのデータがすべて“0”と判定したときは、データが“0”でないバンクのデータを読み出して演算結果を格納するバンクに複写し、当該バンクに対応するオールゼロフラグをセットするようにしたものである。これによって、データがすべて“0”のバンクのデータと他のバンクのデータとを加算するときは、一方のバンクからのデータの読み出しおよび演算回路による演算は行なわないで済むので、演算処理およびメモリのアクセス回数が少なくなって消費電力が低減されるとともに、伸張処理における演算時間が短縮される。

【0018】

また、上記メモリ領域に格納されたデータ同士の演算を行なう演算回路を備え、上記メモリ領域に格納された2つのデータを乗算する処理を行なう際に、上記乗算される2つのデータのうち一方に対応する上記フラグが第1の状態にされているときは、演算後のデータが格納されるべきメモリ領域に対応する上記フラグを第1の状態に設定するようにする。

【0019】

より具体的には、あるバンクのデータと他のバンクのデータとの積を演算しようとするときに、オールゼロフラグを見て一方のバンクのデータがすべて“0”と判定したときは、演算結果を格納するバンクへの書込みを省略して対応するオールゼロフラグをセットするようにしたものである。これによって、データがす

べて“0”のバンクのデータと他のバンクのデータとの積を演算するときは、バッファメモリに対するアクセスおよび演算を行なわないで済むので、演算処理およびメモリのアクセス回数が少なくなつて消費電力が低減されるとともに、伸張処理における演算時間が短縮される。

【0020】

さらに、上記メモリ領域に格納されたデータ同士の演算を行なう演算回路と、上記データが格納される複数のメモリ領域を識別するための情報を管理するメモリ管理手段とを備え、上記メモリ領域に格納された2つのデータを加算する処理を行なう際に、上記加算される2つのデータのうち一方に対応する上記フラグが第1の状態にされているときは、上記フラグが第2の状態のメモリ領域の識別情報と演算後のデータが格納されるべきメモリ領域の識別情報とを入れ替えるとともに、該入替えにより演算後のデータが格納されたメモリ領域に対応する上記フラグを第2の状態に設定するように構成する。

【0021】

より具体的には、各バンク毎にバンク番号を割り振っておいて、あるバンクのデータと他のバンクのデータとの和を演算しようとするときに、オールゼロフラグを見て一方のバンクのデータがすべて“0”と判定したときは、演算結果を格納するバンクへの書込みを省略して、演算結果を格納しようとするバンクとデータがオールゼロでない方のバンクのバンク番号を入れかえるとともに演算結果を格納しようとするバンクに対応するオールゼロフラグをリセットするようにしたものである。これによって、あるバンクのデータと他のバンクのデータとの和を演算しようとするときに、実際に演算を行なわなくても同様の演算結果が得られるため、演算処理およびメモリのアクセス回数が少なくなつて消費電力が低減されるとともに、伸張処理における演算時間が短縮される。

【0022】

さらに、オールゼロフラグがセット状態にされているバンクからデータを読み出す際に、当該バンクのデータに代えて論理“0”のデータを出力する固定データ出力手段を設けるようにした。これにより、バンクのデータを用いた演算においてオールゼロフラグがセット状態にされているバンクの“0”のデータが必要

な場合に、バンクの書込みや読出しを行わずに固定データ出力手段から代わりのデータを得ることができるため、正しい演算結果が容易に得られる。

【0023】

【発明の実施の形態】

以下、本発明の実施例を、図面を用いて説明する。

【0024】

図1は、MPEG1オーディオ規格に従ってオーディオデータを伸張し音を再生する例えばMP3プレーヤのようなオーディオデータ再生システムのブロック図であって、101はネットワークなどの伝送媒体より受信もしくはメモリカードなどの記憶媒体から読み出された圧縮データとしてのビットストリームを取り込む入出力回路、102は取り込まれたビットストリームを格納するフラッシュメモリ、103は操作ボタンなどからの信号を取り込む入出力ポート、104は液晶パネルのような表示器105を制御して表示を行なわせる液晶コントローラ、106はフラッシュメモリ102に格納されたビットストリームを伸張する復号部、107はプログラムに従ってシステム全体を制御する中央処理ユニット（CPU）、108はCPU107が実行するプログラムや固定データを記憶する読出し専用メモリ（ROM）、109はCPU106の作業領域を提供するランダム・アクセス・メモリ（RAM）であり、これらはシステムバス110によって互いに接続されている。

【0025】

さらに、上記復号部106は、フラッシュメモリ102からDMA（ダイレクト・メモリ・アクセス）転送でデータを取り込むDMAコントローラ601、取り込まれたビットストリームのフォーマットを解析し圧縮データを復号処理するビットストリーム・プロセッサ602、復号されたデータや演算処理されたデータが格納されるRAMなどからなるバンクメモリ603、乗算器や加算器、パレルシフタなどからなり逆量子化などの演算を行なうデータ演算ユニット604、デコード（伸張）が終了し出力されるPCMデータが格納されるバッファ605、デコーダ600全体の制御および周波数－時間変換などの演算を行なう制御回路606などから構成される。バッファ605に格納された復号オーディオデー

タは、DA変換器200に供給されてアナログ信号に変換され、ヘッドホンやイヤホンあるいはスピーカなどの音声出力手段300により再生される。なお、ビットストリーム・プロセッサ602の機能は、ソフトウェアによっても実現することができ、その場合には制御回路606がそのような機能を代行することとなる。

【0026】

図1のオーディオデータ再生システムにおいては、ネットワークなどの伝送媒体より受信もしくはメモリカードなどの記憶媒体から読み出されたビットストリーム（圧縮データ）は入出力回路101により取り込まれて、一旦フラッシュメモリ102に格納される。そして、フラッシュメモリ102に格納されたビットストリームは、復号部106内のDMAコントローラ601によりビットストリーム・プロセッサ602へDAM転送され、このビットストリーム・プロセッサ602によりフォーマットが解析されて復号され、所定のバンクメモリ603に格納される。

【0027】

バンクメモリ603に格納されたデータは、データ演算ユニット604と制御回路606の協同により逆量子化、バタフライ演算（フーリエ変換）、逆MDCT演算などの処理がなされてPCM音源データとしてバッファ605に格納され、このバッファ605からDA変換器200へ出力されて、DA変換器200が例えば44.1kHzのようなサンプリング周波数に従ってDA変換を行なって音声出力手段300により再生される。なお、上記逆量子化、バタフライ演算、逆MDCT演算などの処理のうち最後の演算を除く演算の結果は再度バンクメモリ603に格納され、最後の演算処理結果がバッファ605に格納される。

【0028】

ここで、上記復号部106におけるビットストリームからPCMデータを生成するまでの手順を、図2を用いて詳細に説明する。

【0029】

フラッシュメモリから転送されてきたビットストリームは、先ずフレーム単位でそのフォーマットが解析される（ステップS11）。そして、フォーマットか

らビットストリーム内にあるハフマン符号などの符号表（ハフマンテーブル）と量子化で用いられた量子化スケールに対応した逆量子化テーブルをそれぞれ抽出して復号する（ステップS12, S13）。次に、ビットストリームに含まれている可変長のサブバンドサンプルデータを抽出して上記ステップS12で得られたハフマン符号表を用いて復号を行ない、各周波数帯の固定長のサンプルデータ $SPL1, SPL2 \dots SPL32 (= SPLij)$ を得る（ステップS14）。ここまでの処理は、図1のシステムではビットストリーム・プロセッサ602により行なわれ、得られたハフマン符号表、逆量子化テーブルおよびサンプルデータはバンクメモリ603に格納される。

【0030】

次に、ステップS14で復号されたサンプルデータを、ステップS13で得られた逆量子化テーブルを用いて逆量子化すなわちサンプルデータ $SPLij$ と逆量子化係数 $SFij$ との乗算を行ない、演算結果 $Y1, Y2 \dots Yn (= Ymn)$ はバンクメモリ603に格納する（ステップS15）。続いて、折返し歪の低減のためバタフライ演算と呼ばれる演算すなわち同一バンク内のデータの加算または減算 $Bij = Ymn \pm Yml$ を所定の手順で行ない、演算結果 $B1, B2 \dots Bn (= Bij)$ をバンクメモリ603に格納する（ステップS16）。それから逆変形離散コサイン変換（IMDCT）を行なう（ステップS17）。

【0031】

ここで、IMDCT変換は、ステップS16のバタフライ演算結果の Bjk と IMDCT係数 Mik との積和演算 $Cij = \sum Mik * Bij$ を行なうことである。演算結果 $C1, C2 \dots Cn (= Cij)$ はバンクメモリ603に格納される。なお、上記IMDCT係数は予めバンクメモリ603内に格納しておいても良いし、ビットストリームに含ませて伝送するようにしてもよい。なお、上記ステップS16のバタフライ演算とステップS17のIMDCT変換が、図17における周波数-時間変換の逆変換に相当する処理である。

【0032】

上記IMDCT変換終了後に、復号、伸張された上記各周波数帯のサンプルデータ Cij にそれぞれ重み係数を掛けた後、加算して元の1ブロック（8m秒）

のデータを復元するサブバンドデータの合成 ($D = N1 * CiJ + N2 * Ckj$) を行なう (ステップ S18)。そして、合成されたデータは PCM 音源データとしてバッファ 605 に順次格納される。なお、上記演算は一例であって、復号するデータの種類や性質等に応じて演算が一部省略されたり、他の演算と置き換えられたり、別の演算が付加されることもあり、本発明はそのような場合にも適用することができる。

【0033】

図3は、本発明の一実施例を示すもので、図1のオーディオデータ再生システムにおける復号部106の一部をより詳細に示しており、実質的に同一のシステムである。図3において、図1と同一もしくは類似の機能を有する回路ブロックには同一の符号を付して重複した説明は省略する。図3において、符号610で示されているロード/ストア・ユニットは、図1におけるDMAコントローラ601とバッファ605の機能を1つのブロックとして表わしたもので、それらと同一の機能を有する。ビットストリーム・プロセッサ602が破線で示されているのは、このプロセッサの機能は制御回路606により実現することも可能であることを表わしている。

【0034】

制御回路606は、例えばマイクロ命令を格納したマイクロROMやマイクロ命令をデコードして制御信号を生成する命令デコーダ、実行する命令の順序を制御するシーケンサなどからなる制御部CNTと、バンクメモリ603の各バンクの番号を記憶したバンクIDテーブルBITを備えバンク番号を管理するバンク管理ユニットBMUと、バンクメモリ603の各バンクをアクセスするためのアドレスを演算するアドレス演算ユニットACUなどから構成される。データ演算ユニット604は、論理演算を行なう論理演算ユニットALUと、積和演算を行なうMACと、データのビットシフトなどの処理を行なうバレルシフタBSFなどから構成される。

【0035】

また、本実施例においては、バンクメモリ603が例えば36個のようなバンクBNK0、BNK1……BNK35により構成されている。1つ1つのバンク

BNK i ($i = 0, 1, \dots, 35$) は、例えば1ブロックの256個のようなサンプルデータを保持可能な 256×32 ビットのような記憶容量を有するように構成されるとともに、自己に与えられたバンク番号が入力されるとバンクを選択状態にするバンク活性化回路BAC i と、バンク内の実際のデータにかかわらずそれがオールゼロであることを宣言するためのオールゼロフラグAZF i とが設けられている。前述の復号、伸張処理の過程で演算されるサンプルデータや係数はそれぞれ異なるバンクに格納され、データ演算ユニット604で演算された後、元のバンクあるいはさらに別のバンクに格納される。

【0036】

なお、図3において、ADBはアドレス演算ユニットACUで生成されたアドレスをバンクメモリ603の各バンクBNK0～BNK35に供給するアドレスバス、DTBはロード/ストア・ユニット610とバンクメモリ603とデータ演算ユニット604との間でデータを転送可能にするデータバスである。

【0037】

各バンクがオールゼロフラグAZF i を有することと、このオールゼロフラグAZF i の利用の仕方が本発明の大きなポイントである。JPEG規格やMPEG規格による画像データの圧縮やオーディオデータの圧縮に使用されるDCT変換では、DCT変換後のDCT係数に偏りがあり特定の周波数成分が多く現われる傾向があるため、これを利用してDCT係数を量子化したときに「0」となる周波数成分が非常に多くなり、これによって大きな圧縮率が得られる。ここで、「0」となる周波数成分が多いにもかかわらずこれをいちいちバンクメモリに書き込んでいたのでは、「0」の書込みに要する時間のメモリのアクセス時間に占める割合が非常多くなってしまい、いかにも無駄である。本発明がバンクメモリの各バンクにオールゼロフラグAZF i を設けたのは、かかるバンクメモリへの「0」の書込みおよび読出しを省略するためである。

【0038】

以下に、上記オールゼロフラグAZF i を利用したバンク制御およびデータの復号、伸張処理について説明する。

【0039】

仮に、4つの周波数帯のデータ $\{A_0, A_1, \dots, A_n\}$, $\{B_0, B_1, \dots, B_n\}$, $\{C_0, C_1, \dots, C_n\}$, $\{D_0, D_1, \dots, D_n\}$ がビットストリームにのって入ってきて、これを4つのバンク B_{NK0} , B_{NK1} , B_{NK2} , B_{NK3} に書き込む場合を考えると、バンクメモリにオールゼロフラグ AZF_i が設けられていないシステムにおいては、 $C_0 \sim C_n$ と $D_0 \sim D_n$ が全て「0」であったとしても、各周波数帯のデータのバンクメモリへの書込みは、図4に示すようにデータが「0」か否かにかかわらずすべてのバンクに対して行なわれることとなる。なお、図において、 A_0, A_1, \dots, A_n , B_0, B_1, \dots はそれぞれが32ビットのようなビット長のデータであり、また n は例えば255である。

【0040】

これに対し、本発明の実施例においては、上記のような4つのデータをバンク B_{NK0} , B_{NK1} , B_{NK2} , B_{NK3} に書き込む場合、図5に示すように、バンク B_{NK0} と B_{NK1} にはデータ $\{A_0, A_1, \dots, A_n\}$, $\{B_0, B_1, \dots, B_n\}$ を書き込むが、バンク B_{NK2} と B_{NK3} へのデータの書込みは行なわずに、元のデータ $\{c_0, c_1, \dots, c_n\}$, $\{d_0, d_1, \dots, d_n\}$ を残しておいて、対応するオールゼロフラグ AZF_2 , AZF_3 に「1」をセットしてデータがすべて「0」であることを宣言させるものである。また、各バンクからデータを読み出す際には、まずオールゼロフラグ AZF_i の状態を読み込んでそれが「1」である場合にはバンクからのデータの読出しは行なわないようにする。

【0041】

次に、本発明の実施例のように各バンクにオールゼロフラグ AZF_i が設けられているシステムにおいて、各バンクに格納されているデータ同士の具体的な演算の仕方について説明する。例えば図6(A)に示すように、オールゼロフラグ AZF が「1」になっているバンク B_{NK0} のオールゼロのデータと、オールゼロフラグ AZF が「0」になっているバンク B_{NK1} の有為なデータとを掛算してバンク B_{NK2} に格納する場合を考える。ここで、バンク B_{NK0} のデータとバンク B_{NK1} のデータとの掛算（乗算）とは、バンク B_{NK0} の i ($i=0 \sim$

n) 番目のデータ a_i とバンク $BNK1$ の i 番目のデータ b_i との掛算を $i=0$ から $i=n$ まで順次行なうことを意味する。

【0042】

この場合、本実施例のシステムにおいては、実際のデータの掛算は行なわないで、図6(B)のようにバンク $BNK2$ のオールゼロフラグ $AZF2$ に「1」をセットして終了する。このように、オールゼロフラグ AZF を利用した場合、バンクデータ同士の掛算において一方がオールゼロのデータであるときには、バンクからのデータの読出しも書込みも一切行なう必要がない。なお、図6において、オールゼロフラグ AZF の欄に「X」に記載されているのは、「0」または「1」のいずれでもよいことを意味している。

【0043】

次に、例えば図7(A)に示すように、オールゼロフラグ AZF が「1」になっているバンク $BNK0$ のオールゼロのデータと、オールゼロフラグ AZF が「0」になっているバンク $BNK1$ の有為なデータとを加算してバンク $BNK2$ に格納する場合を考える。ここで、バンク $BNK0$ のデータとバンク $BNK1$ のデータとの加算とは、バンク $BNK0$ の i ($i=0\sim n$) 番目のデータ a_i とバンク $BNK1$ の i 番目のデータ b_i との加算を $i=0$ から $i=n$ まで順次行なうことを意味する。

【0044】

オールゼロフラグがない場合における加算では、バンク $BNK1$ のデータはもちろんバンク $BNK0$ のデータも読み出してそれぞれのデータを加算してバンク $BNK2$ に書き込む処理が必要である。これに対して、本実施例のシステムにおいては、実際のデータの加算は行なわないで、図7(B)のようにバンク $BNK1$ のデータをバンク $BNK2$ に複写、すにわちバンク $BNK1$ からデータを読み出してバンク $BNK2$ に書き込むとともに、 $BNK2$ のオールゼロフラグ $AZF2$ を「0」にリセットして終了する。このように、オールゼロフラグ AZF を利用した場合、バンクデータ同士の加算において一方がオールゼロのデータであるときには、1つのバンクからのデータの読出しと他のバンクへの書込み動作のみ行ない、実際の演算は省略することができる。

【0045】

さらに、上記の場合に加算後のデータをバンクBNK2ではなくBNK1に戻す場合には、バンクの読出し、書込みおよび演算を一切行なう必要がない。同様に、オールゼロフラグAZFが「1」になっているバンクのデータ同士を加算する場合にも、実際の加算は行なわずに加算結果を入れるバンクに対応して設けられているオールゼロフラグAZFに「1」をセットするだけで良い。

【0046】

また、図3のシステムと類似の構成を有し、各バンクBNK0～BNK35に与えられているバンク番号が固定ではなく、バンク管理ユニットBMUによってバンク番号が任意に付け替え可能に構成されているシステムにおいては、オールゼロフラグAZFが「1」になっているバンクBNK0のオールゼロのデータとオールゼロフラグAZFが「0」になっているバンクBNK1の有為なデータを加算してバンクBNK2に格納する場合には、図8のように、バンクBNK1とBNK2のバンク番号を入れ替えるとともに、新たなバンクBNK1（元のバンクBNK2）のオールゼロフラグAZFを「0」にリセットして終了することができる。このように、バンク番号が入れ替え可能で、一方がオールゼロのデータである場合には、バンク番号の入れ替えとオールゼロフラグAZFのリセットのみ行なえば良く、バンクの読出し、書込みおよび演算を一切行なう必要がない。

【0047】

なお、上記の場合、番号が入れ替えられた新たなバンクBNK1（3列目）は、有為でないデータが格納されているにもかかわらずオールゼロフラグAZFが「0」にされることとなるが、このデータが有為でないことはバンク管理ユニットBMUにおいて把握することが可能であるので、何ら問題はない。また、バンク管理ユニットBMUにおけるバンクの管理が充分になされていれば、上記バンク番号の入れ替えの際に、新たなバンクBNK1（元のバンクBNK2）のオールゼロフラグAZFを「0」にリセットする操作を省略することが可能である。具体的には、各バンクに対応して上記オールゼロフラグAZF以外に例えばバンク内のデータが有効か無効かを示すフラグを設けて、バンク管理ユニットBMUにより管理することが考えられる。

【 0 0 4 8 】

次に、オールゼロフラグを利用して上記のようなバンク制御を可能にする具体的なシステムの構成例を、図 9 および図 1 0 を用いて説明する。このうち、図 9 はバンク番号が固定方式のシステムの例を、また図 1 0 はバンク番号が可変であるシステムの例をそれぞれ示す。

【 0 0 4 9 】

図 9 のバンク番号が固定方式のシステムにおいては、例えばアドレス演算ユニット A C U から出力されるアドレスのバンク番号に相当するアドレスの上位 6 ビットをデコードして対象バンクを指定するイネーブル信号 B E N 0 ~ B E N 3 5 を生成して各バンク B N K 0 ~ B N K 3 5 に供給するバンク番号デコーダ B N D が設けられる。また、バンク B N K 0 ~ B N K 3 5 の出力側にはセクタ S E L が設けられ、このセクタ S E L の一方の入力端子には各バンクのデータ出力端子が共通に接続されているとともに、他方の入力端子には固定データ“0”が入力されるように構成されている。

【 0 0 5 0 】

この実施例のシステムでは、アドレス演算ユニット A C U から出力されるアドレスは各バンク B N K 0 ~ B N K 3 5 に共通に供給され、イネーブル信号 B E N が有効レベルにされたバンクのみが活性化されて、アドレスの下位 8 ビットにより指定される番地のデータを読み出したり、書込みが行なわれたりする。

【 0 0 5 1 】

また、このシステムでは、バンク管理ユニット B M U が各バンクのオールゼロフラグ A Z F を管理する機能を備えており、データの書込みや加算、乗算の際に、前述したようにオールゼロのデータを書き込むバンクに関してはフラグ A Z F に「1」をセットすることで書込みを省略する。また、データの読出しの際には、バンク管理ユニット B M U が先ず対象バンクのオールゼロフラグ A Z F をチェックしてそれが「1」になっているときはバンクからのデータの読出し動作は行なわずに、バンクの出力側に設けられているセクタ S E L を固定データ“0”側に切り替える。これによって、復号処理が終了してサブバンドサンプルデータの合成を行なう際など、オールゼロフラグ A Z F が「1」になっているバンクか

らデータを読み出すべきときに、バンクのデータに代えて固定データ“0”がセレクトA SELから出力されて、正しい復号データの読出しと、それを用いた演算が行なわれることとなる。

【0052】

図10のシステムにおいては、バンク管理ユニットBMUにバンク番号を管理するためのバンクIDテーブルBITが設けられる。また、各バンクBNK0～BNK35にはそれぞれオールゼロフラグAZF0～AZF35以外に、自己のバンク番号を保持するレジスタとこのレジスタに保持されているバンク番号とアドレス演算ユニットACUから供給されるアドレス上位のバンク番号とを比較する比較器とからなるバンクアクセス判定回路BAJ0～BAJ35が設けられている。

【0053】

また、各バンクの入力側には入力アドレスと入力データを通過させたり遮断させたりできるゲート手段TGT0～TGT35が設けられており、これらのゲート手段TGT0～TGT35はそれぞれ上記バンクアクセス判定回路BAJ0～BAJ35からの信号によって、アドレス演算ユニットACUから出力された例えばアドレスの上位6ビットのバンクアドレスがバンクアクセス判定回路BAJ0～BAJ35内のバンク番号レジスタ内のバンク番号と一致したときに通過状態となるように制御される。

【0054】

さらに、各バンクの出力側には各バンクBNK0～BNK35の出力のうち1つを選択して後段へ伝達可能にするマルチプレクサMUXと、バンク出力データに代えて固定データ“0”を出力可能なセレクトA SELとが設けられている。上記セレクトA SELは前記実施例と同様に、バンクデータの読出し時にバンク管理ユニットBMUが、各バンクBNK0～BNK35のオールゼロフラグAZF0～AZF35をチェックしてフラグの状態に応じて切り替えられる。また、マルチプレクサMUXもバンク管理ユニットBMUによって、バンクIDテーブルBITを参照して所望のバンクのデータが出力されるように制御される。

【0055】

さらに、この実施例のシステムにおいては、上記バンク管理ユニットBMUが、バンクIDテーブルBITを書き換えるとともに、各バンクBNK0～BNK35に設けられているバンクアクセス判定回路BAJ0～BAJ35内のバンク番号レジスタを書き換えることができるように構成されている。これによって、例えば図8を用いて説明したようなバンク番号の付け替えおよびオールゼロフラグAZFのセットを行なうことにより、オールゼロフラグAZFが「1」になっているバンクのデータとオールゼロフラグAZFが「0」になっているバンクのデータを加算したのと同じ結果を、バンクからのデータの読出しおよび演算とバンクへの書込みを行なうことなく得ることができる。

【0056】

なお、図10のシステムにおいては、バンクアクセス判定回路BAJ0～BAJ35を設けてハードウェアでバンク番号の入替えを行なっているが、バンク番号の入替えはバンク管理ユニットBMUまたは制御部CNTがソフトウェアで行なうように構成することも可能である。また、バンク番号を設定するレジスタを有するバンクアクセス判定回路BAJ0～BAJ35を設ける代わりに、各バンクにそれぞれバンク管理ユニットBMUから出力されるバンク番号をデコードするデコーダを設けて、バンク管理ユニットBMUがバンクIDテーブルBITを参照して所望のバンクを指定するバンク番号を出力するように構成することも可能である。

【0057】

次に、上記バンク管理ユニットBMUによるバンク管理を含めた復号部106におけるオーディオデータの復号の手順を、図11を用いて説明する。

【0058】

復号部106が復号処理を開始すると、先ずバンク番号を示す変数*i*を「0」にする（ステップS21）。それから、バンク番号*i*のバンクへの書込みを開始して良いか判定する（ステップS22）。この判定は、具体的には、前フレームの復号が終了して次のフレームの復号を開始して良いかを、例えば制御回路606内の制御状態を示すフラグやステータスレジスタを参照することで行なえる。そして、書込み開始可能と判定すると、次のステップS23へ進んでビットスト

リーム・プロセッサ602によりヘッダ部のデコードを行なわせ、フレーム内のデータ数を取得する。

【0059】

次に、ビットストリームからサンプルデータを抽出してハフマン復号化を行なう（ステップS24）。そして、復号された各周波数帯のサンプルデータをステップS21で設定したバンク番号 i （ $=0$ ）のバンクに格納し、当該バンクに対応するオールゼロフラグAZFiを「0」にリセットする（ステップS25、S26）。それから、復号したサンプルデータがバンクの容量に達したならば、バンク番号 i をインクリメント（ $i \leftarrow i + 1$ ）する（ステップS27）。

【0060】

その後、フレーム内の全有効データすなわち「0」でないデータの復号が終了したか判定する（ステップS28）。この有効データ数の判定は、予めビットストリーム内に有効データ数情報が付加されて送信されてくるので、復号したデータの数を計数して受信した有効データ数情報のデータ数と一致したか否かを判定することで行なえる。そして、全有効データの復号が終了していないと判定したときはステップS24へ戻って次のサンプルデータの復号化を行ない、終了したと判定すると、次のステップS29で現バンクの残りの領域に全てデータ「0」を書き込む。

【0061】

しかる後、バンク番号 i をインクリメント（ $i \leftarrow i + 1$ ）して、当該バンクのオールゼロフラグAZFiを「1」にセットする（ステップS30、S31）。これは、MPEG1規格によるオーディオデータの圧縮の性質から、各周波数帯のサンプルデータに偏りがあり、サンプルデータに「0」が現われると残りのサンプルデータはすべて「0」になる特徴があるので、「0」のサンプルデータを次のバンクに書き込む代わりに、オールゼロフラグAZFiを「1」にセットすることでバンクへの「0」の書込みを省略するためである。

【0062】

それから、すべてのバンクへのデータの書込み（オールゼロフラグAZFiのセットを含む）が終了したか判定する（ステップS32）。そして、終了してい

なければステップ S 3 0 へ戻って上記処理を繰り返し、ステップ S 3 2 で「終了」と判定した時点で、残りの復号処理（バタフライ演算処理、逆 M D C T 変換処理およびサブバンド合成処理）S 3 3 を行なって、ステップ S 3 4 ですべてのフレームについて復号が終了したか判定し、終了していないときはステップ S 2 1 へ戻って上記処理を繰り返す。

【 0 0 6 3 】

次に、本発明を J P E G 規格に従って圧縮された画像データを伸張する画像データ再生システムに適用した場合の実施例を説明する。

【 0 0 6 4 】

先ず、J P E G に従った画像データの圧縮（符号化）と伸張（復号化）の手順を、図 1 2 を用いて簡単に説明しておく。J P E G 規格の圧縮では、圧縮するデジタル原画像 4 0 0 が 8×8 の画素のブロック B 1, B 2 ……に分割され、各ブロック毎に 3 原色の各色ごとのサンプルデータ R G B が輝度 Y と色差成分 C b, C r のデータに変換される。そして、Y, C b, C r のそれぞれのデータに対して 2 次元 D C T 変換（空間一周波数変換）が行なわれてから、適当な量子化テーブル 4 1 0 を使用した量子化（割算と四捨五入）によるデータ圧縮が行なわれる。その後、ハフマン符号表 4 2 0 を用いてハフマン符号化処理が行われてさらにデータが圧縮される。

【 0 0 6 5 】

しかる後、符号化されたサブバンドサンプルデータにヘッダやフォーマット情報、エラーチェックコード、ビット割り当て情報、スケールファクタ情報などの情報を付加した所定のフォーマットのフレームを構成する。そして、ブロック毎のフレームが連続したビットストリームが生成されて復号システムに送られて来る。

【 0 0 6 6 】

画像データの復号は、図 1 7 に示すオーディオデータの復号と処理内容が若干異なるのみで手順はほぼ同じである。すなわち、まず、ビットストリームを解析してサンプルデータを抽出して符号表 4 2 0 を用いたハフマン復号化を行ない、次に量子化テーブル 4 1 0' を用いた逆量子化（掛算）を行なう。サンプルデー

タは、オーディオデータでは8m秒のデータを1ブロックとしてそれを32の周波数帯に分割して符号化したものであるのに対し、画像データでは8×8画素のデータを1ブロックとして色成分毎に空間-周波数変換（RGB→YCbCr変換およびDCT変換）と量子化を行なってからハフマン符号化したものである。画像データの復号では、上記逆量子化の後、周波数-空間変換に相当する逆DCT変換とYCbCr→RGB変換が行なわれて再生画像400'が生成される。

【0067】

画像データもオーディオデータと同様に空間周波数で表わすと周波数成分に偏りがあり、通常の画像データは低周波数が多く高周波数成分は少ないという特徴があるため、ハフマン復号化したサンプルデータは例えば図13（A）に示すように、有効なデータは左上の低周波数成分に多く現われ値も大きい、右下の高周波数成分のDCT係数はほとんどが「0」となる。なお、JPEG規格による圧縮では、各ブロックの64個の周波数成分は図13（B）に矢印で示すような順番で符号化され、この順番でビットストリームに載せて送られて来る。

【0068】

本実施例の画像データの再生システムにおいても、この画像データの性質を利用して、各周波数帯のデータを格納するバンクメモリの各バンクにオールゼロフラグを設けておいて、復号化処理における演算の際に各バンクへの「0」の書込みや「0」のデータとの乗算などの演算をオールゼロフラグのセット/リセット処理で置き換えることによって処理の簡略化を図れるようにしている。

【0069】

図14には、本発明をJPEG規格に従って圧縮された画像データを伸張する画像データ再生システムに適用した場合の実施例を示す。

【0070】

図14において、501は入力されたビットストリームを解析しハフマン復号化を行なうビットストリームデコーダ、502はオールゼロフラグAZFを有し8×8の64個の周波数帯のデータを図15にG0～G7ですのような8つのグループに分けてそれぞれ記憶する8個のバンクBNK0～BNK7からなるバンクメモリ、503は積和演算を行なう演算回路、504は量子化テーブルやDCT

係数テーブルを記憶する係数メモリ、505は逆DCT変換に用いられる1ブロック分のデータを保持するバッファメモリ、506は逆DCT変換後のデータを保持するフレームバッファ、507はYCbCr→RGB変換を行なう変換回路、508はシステム全体を制御する制御回路である。

【0071】

なお、上記係数メモリ504はバンクメモリ502と同じRAM内に別のバンクとして用意しておいても良い。逆DCT変換用バッファ505も同様である。ただし、オールゼロフラグが必要なのは、最初の8個のバンクBNK0～BNK7に対してのみで、他のバンクがあってもそれらに関してはオールゼロフラグは不用である。

【0072】

また、SEL1～SEL4はそれぞれデータを選択するためのセレクトタである。このうち、セレクトタSEL2は一方の入力端子に各バンクのデータ出力端子が共通に接続されているとともに、他方の入力端子には固定データ“0”が入力され、バンクからのデータ読出し時にオールゼロフラグAZFの状態に応じてセレクトタが切り替えられて読出しデータの代わりに固定データ“0”を出力できるように構成されている。

【0073】

次に、図14の画像データ再生システムにおける具体的に復号処理を、図16を用いて説明する。図16は画像データ再生システムにおける復号処理の流れを図式化したものである。

【0074】

再生システムに入力されたビットストリームはビットストリームデコーダ501によって解析され、入力ビットストリームから抽出された量子化テーブルが係数メモリ505に格納されるとともに、ハフマン復号化された64個の周波数帯のサンプルデータは8個のバンクメモリBNK0～BNK7に順次格納される。具体的には、各周波数帯のサンプルデータは図16の(b)に矢印で示すようにジグザグに格納される。このとき、バンクメモリへ書き込むべき周波数帯のデータがすべて「0」のときはバンクメモリへの書込みを行わずに、対応するオー

ルゼロフラグを「1」にセットする。

【0075】

一般には、圧縮側は圧縮過程で途中から終わりまでデータ「0」が連続するようになるとデータ「0」を送って来ないので、図16(b)の順序でデータを埋めていくと復号データが64個に満たなくなる。その場合、復号システム側で残りのデータを「0」で埋めてバンクメモリ502に格納してやる必要があるが、この実施例のシステムではバンクメモリへの書込みを行わずに、対応するオールゼロフラグが「1」にセットされる。このようにして、サンプルデータをバンクメモリへ格納する際に実際のメモリへのアクセス回数が削減されることによって消費電流が大幅に減少されることとなる。

【0076】

次に、バンクメモリ502から各周波数帯のサンプルデータを読み出して逆量子化を行なって再びバンクメモリ502へ格納するが、このときオールゼロフラグAZFをチェックして「0」の立っているデータに関してはデータを読み出して量子化テーブルの値との掛算を行なう。そして、オールゼロフラグが「1」のデータに関してはデータの読出しおよび掛算は省略してオールゼロフラグを「1」のままにする。逆量子化は、逆量子化される前の i 行 j 列のサンプルデータを X_{ij} 、逆量子化テーブルの値を C_{ij} とすると、式 $Y_{ij} = X_{ij} \times C_{ij}$ で表わされる。従って、データが「0」のところは逆量子化後も「0」のままであるので、上記のように演算を省略することができる。これによって、逆量子化におけるバンクメモリへのアクセスおよび演算に要する時間が大幅に短縮されることとなる。

【0077】

次の逆DCT変換は、2段階の行列演算によって行なわれる。まず1段階では、図16(e)に矢印で示すように列方向でバンクメモリ内のデータ X_j を読み出して、図14の積和演算回路503において、IDCT係数 M_{ij} との行列演算 $Y_j = \sum M_{ij} \times X_j$ を $j=0$ から $j=7$ まで即ち1列目から8列目まで順に行ない、演算結果をバッファメモリ505内に格納する。そして、これを各バンクBNK0～BNK7内の全てのデータに対して行なう。そして、次の2段階で

は、図16(f)に矢印で示すように行方向でバッファメモリ内のデータ X_i を読み出して、IDCT係数 M_{ij} との行列演算を $i=0$ から $i=7$ まで即ち1行目から8行目まで順に行ない、演算結果をフレームバッファ506内に格納する(図16(g))。

【0078】

この実施例の画像データ再生システムにおいては、上記1段目の行列演算の際に、各バンクBNK0～BNK7からデータを読み出すときに先ずオールゼロフラグAZFをチェックして、「1」の立っているデータに関してはバンクからデータを読み出す代わりにセクタSEL2を切り替えて固定データ“0”を積和演算回路503へ渡すように制御される。このようにして、逆DCT変換におけるバンクメモリへのアクセスが省略されることによって消費電流が大幅に減少されることとなる。また、メモリのアクセス時間よりもセクタから固定データ“0”を出力させる時間の方が短くて済むため、トータルの演算所要時間も短縮される。

【0079】

以上本発明者によってなされた発明を実施例に基づき具体的に説明したが、本発明は上記実施例に限定されるものではなく、その要旨を逸脱しない範囲で種々変更可能であることはいうまでもない。例えば、前記実施例においては、オールゼロフラグAZFのセット状態を、対応するバンクのデータがすべて“0”である状態に対応させているが、オールゼロフラグAZFのリセット状態を、対応するバンクのデータがすべて“0”である状態に対応させるようにしてもよい。

【0080】

また、前記実施例のオーディオデータ再生システム(図9、図10参照)においては、各メモリバンクBNK0～BNK35に対応してアドレスのデコーダが設けられている場合を説明したが、バンクメモリ603のアドレスデコーダはメモリ全体に対して共通に設けられ、バンク管理ユニットによるソフトウェア的な処理によりメモリ内が仮想的なバンクに分割されるように構成されていても良い。さらに、本明細書におけるオーディオデータには狭義の音楽データのみならず朗読などの音声データも含まれる。

【 0 0 8 1 】

以上の説明では主として本発明者によってなされた発明をその背景となった利用分野であるMPEG1レイヤIII規格に従ったオーディオデータの再生システムおよびJPEG規格の画像データの再生システムに適用した場合について説明したが、本発明はデジタル圧縮データの伸張システム一般に利用することが可能である。

【 0 0 8 2 】

【発明の効果】

本願において開示される発明のうち代表的なものによって得られる効果を簡単に説明すれば下記のとおりである。

【 0 0 8 3 】

すなわち、本発明に従うと、MPEG規格あるいはJPEG規格など所定の圧縮方式によって圧縮されたデータを伸張するシステムにおいて、RAMへのアクセス回数を減らして、システムの消費電力を低減することができるとともに、復号されたデータに対する演算処理を簡略化して、システムの負担を軽減するとともに演算処理速度を高めることができるという効果がある。

【図面の簡単な説明】

【図1】

本発明を適用したMPEG1オーディオ規格に従ってオーディオデータを伸張し音声を再生するオーディオデータ再生システムの一実施例を示すブロック図である。

【図2】

図1の実施例の復号部におけるビットストリームからPCMデータを生成するまでの手順を示すフローチャートである。

【図3】

図1のオーディオデータ再生システムの復号部をより具体的に示したブロック図である。

【図4】

オールゼロフラグを有していない伸張システムにおけるバンクメモリへのデー

タの書込み例を示す説明図である。

【図 5】

オールゼロフラグを有する実施例の伸張システムにおけるバンクメモリへのデータの書込み例を示す説明図である。

【図 6】

オールゼロフラグを有する実施例の伸張システムにおいて乗算を行なった場合のフラグおよびバンクメモリの格納データの変化の様子を示す説明図である。

【図 7】

オールゼロフラグを有する実施例の伸張システムにおいて加算を行なった場合のフラグおよびバンクメモリの格納データの変化の様子を示す説明図である。

【図 8】

オールゼロフラグおよびバンク番号管理ユニットを有する実施例の伸張システムにおいて加算を行なった場合のフラグとバンク番号およびバンクメモリの格納データの変化の様子を示す説明図である。

【図 9】

オールゼロフラグおよびバンク番号管理ユニットを有する伸張システムの一実施例（バンク番号固定方式）を示すブロック図である。

【図 1 0】

オールゼロフラグおよびバンク番号管理ユニットを有する伸張システムの他の実施例（バンク番号可変方式）を示すブロック図である。

【図 1 1】

バンク管理ユニットによるバンク管理を含めた実施例の伸張システムの復号部におけるオーディオデータの復号の手順を示すフローチャートである。

【図 1 2】

J P E G に従った画像データの圧縮（符号化）と伸張（復号化）の手順の概略を示す説明図である。

【図 1 3】

J P E G 規格により圧縮されたデータをハフマン復号化したサンプルデータの D C T 係数の例と、J P E G 規格における D C T 変換された周波数成分の符号化

の順番を示す説明図である。

【図 1 4】

本発明を J P E G 規格に従って圧縮された画像データを伸張する画像データ再生システムに適用した場合の実施例を示すブロック図である。

【図 1 5】

図 1 4 の実施例の伸張システムにおいてバンクメモリへ格納する際のデータの分割の仕方を示す説明図である。

【図 1 6】

J P E G 方式の伸張システムにおける具体的なデータの処理の手順を示す説明図である。

【図 1 7】

M P E G 1 のレイヤ III に従って圧縮されたオーディオデータを伸張する復号処理の手順の概略を示すフローチャートである。

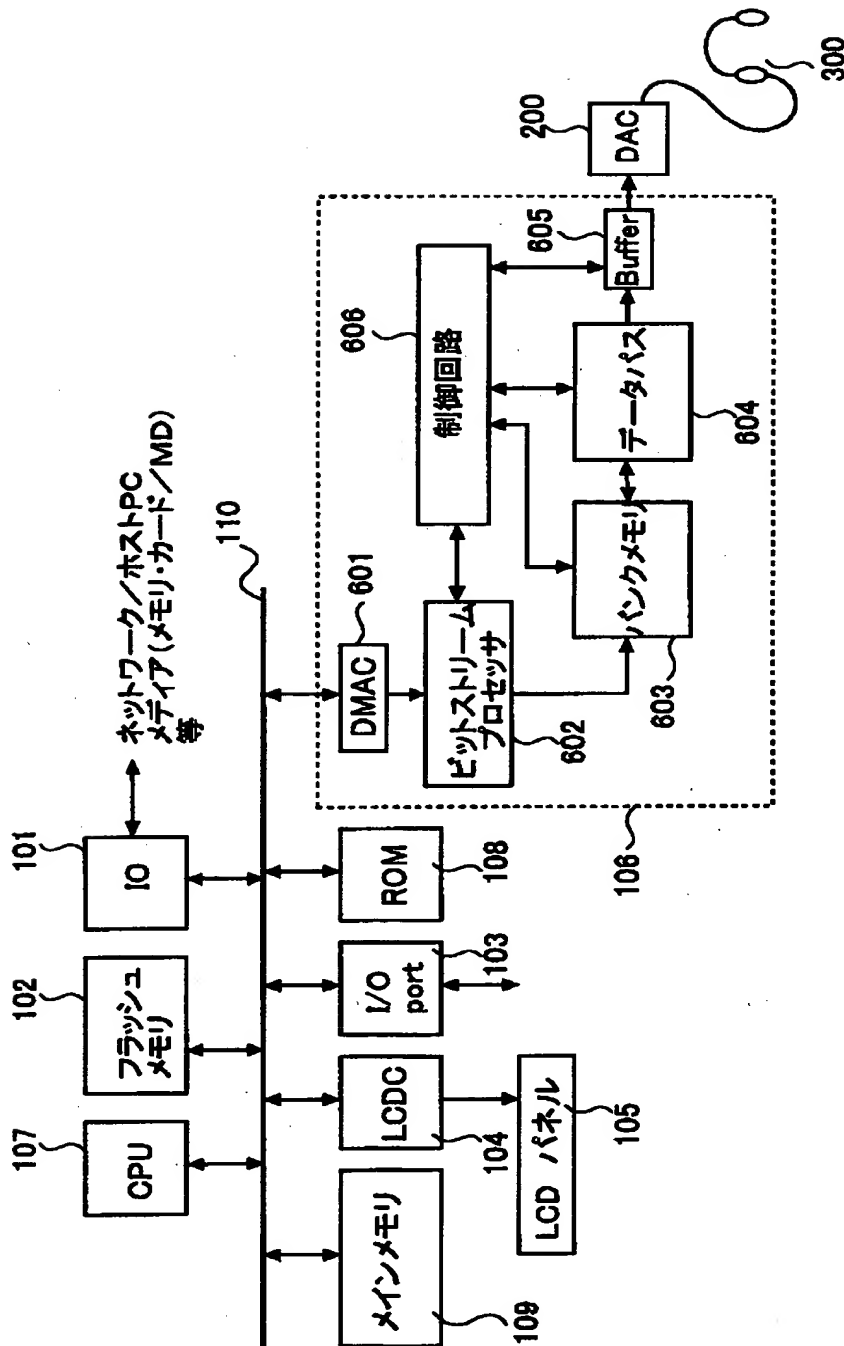
【符号の説明】

- 1 0 1 入出力回路
- 1 0 2 フラッシュメモリ
- 1 0 3 入出力ポート
- 1 0 4 液晶コントローラ
- 1 0 5 表示器
- 1 0 6 復号部
- 1 0 7 C P U
- 1 0 8 読出し専用メモリ (ROM)
- 1 0 9 ランダム・アクセス・メモリ (RAM)
- 1 1 0 システムバス
- 2 0 0 D A 変換器
- 3 0 0 音声出力手段
- 6 0 0 デコーダ (復号システム)
- 6 0 1 D M A コントローラ
- 6 0 2 ビットストリーム・プロセッサ

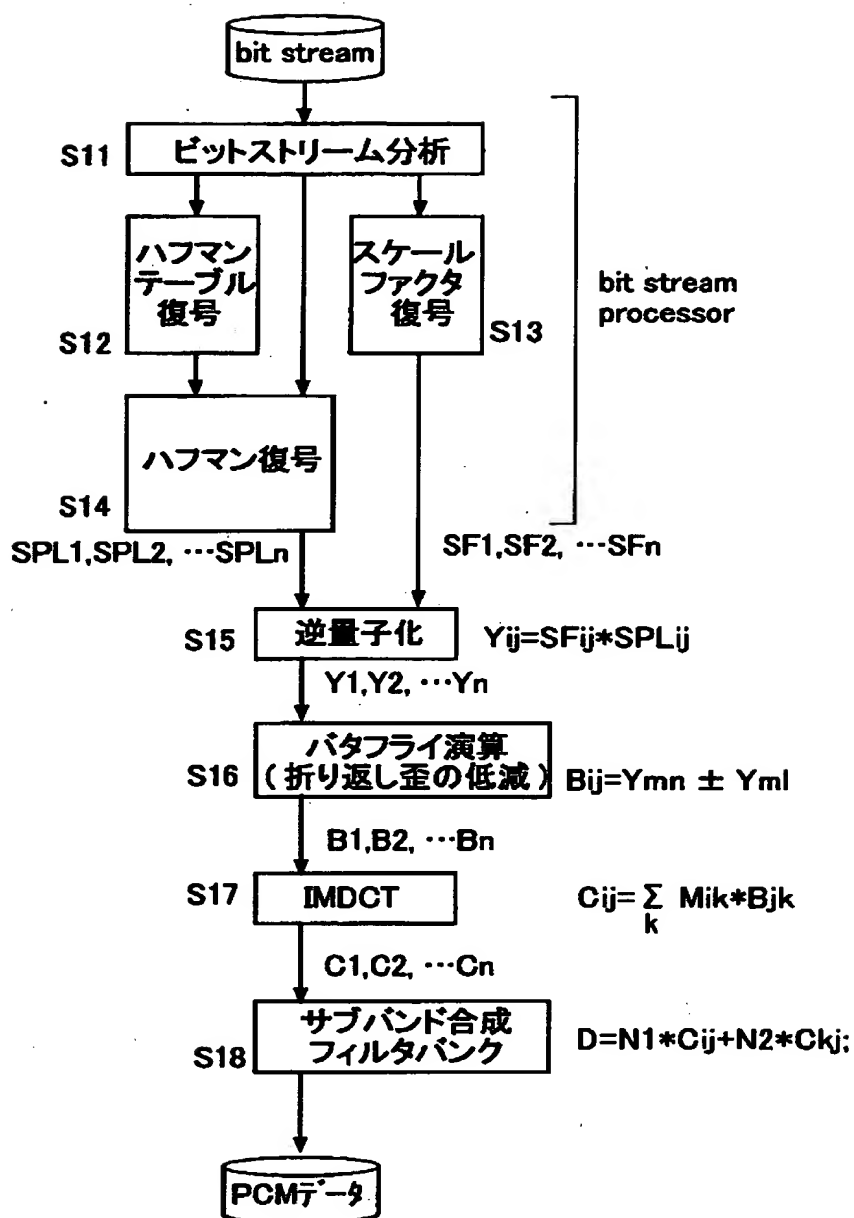
- 603 バンクメモリ
- 604 データ演算ユニット
- 605 バッファ
- 606 制御回路

【書類名】 図面

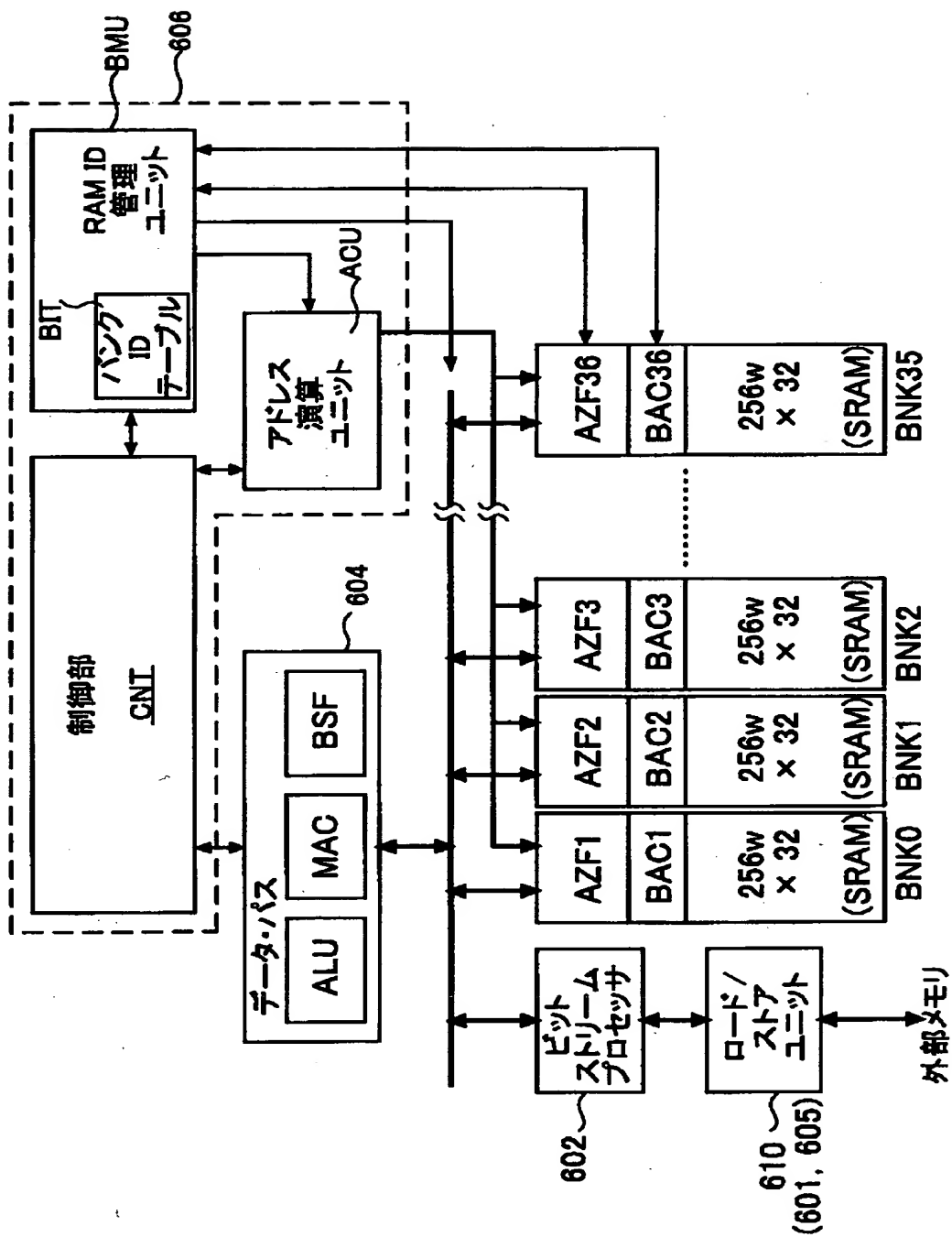
【図 1】



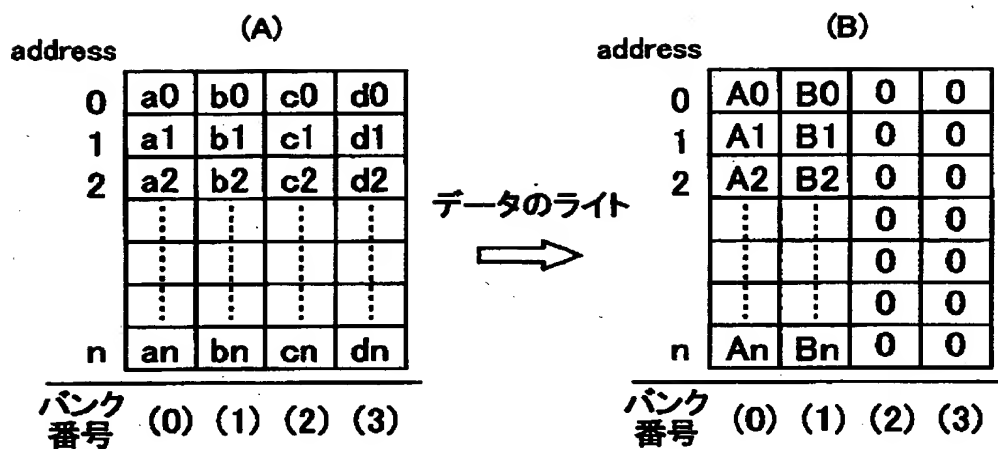
【図 2】



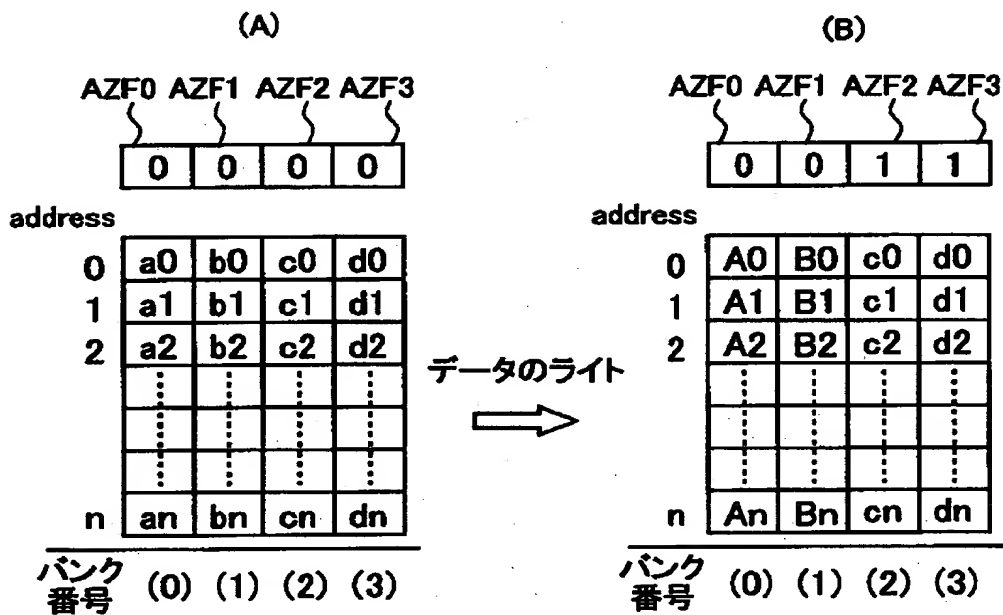
【図 3】



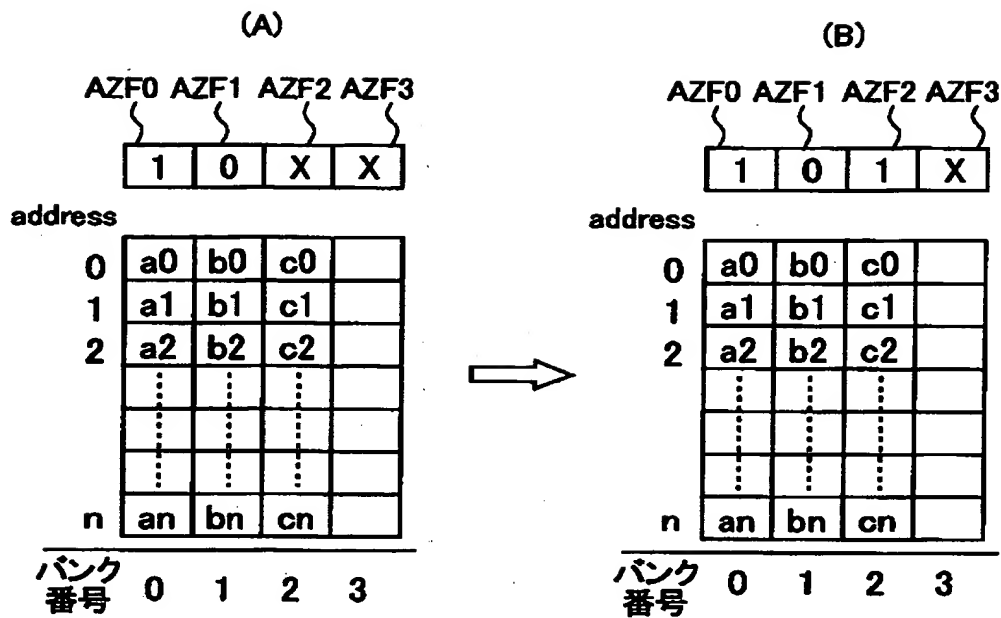
【図 4】



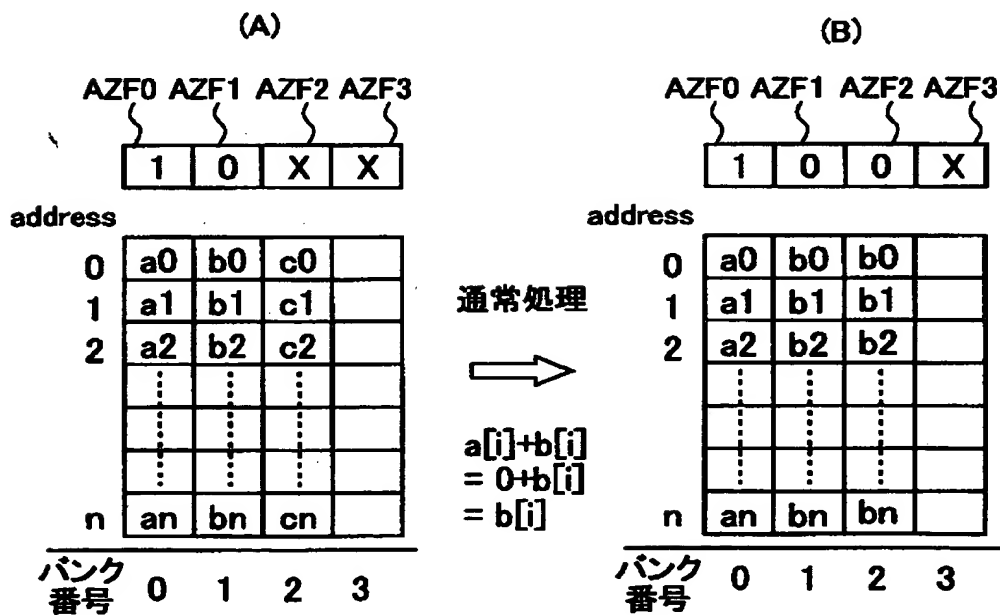
【図 5】



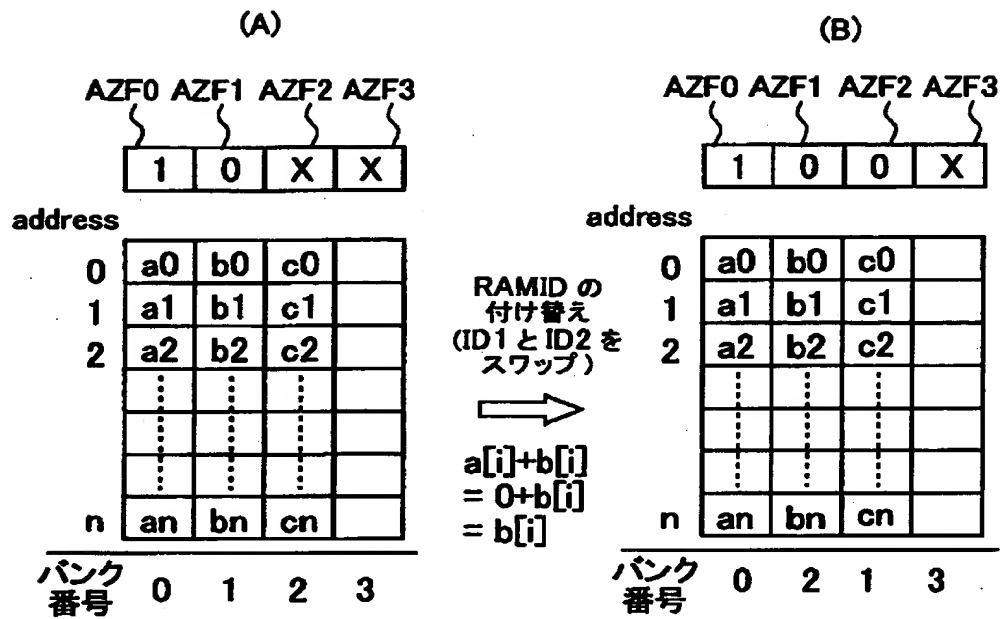
【図 6】



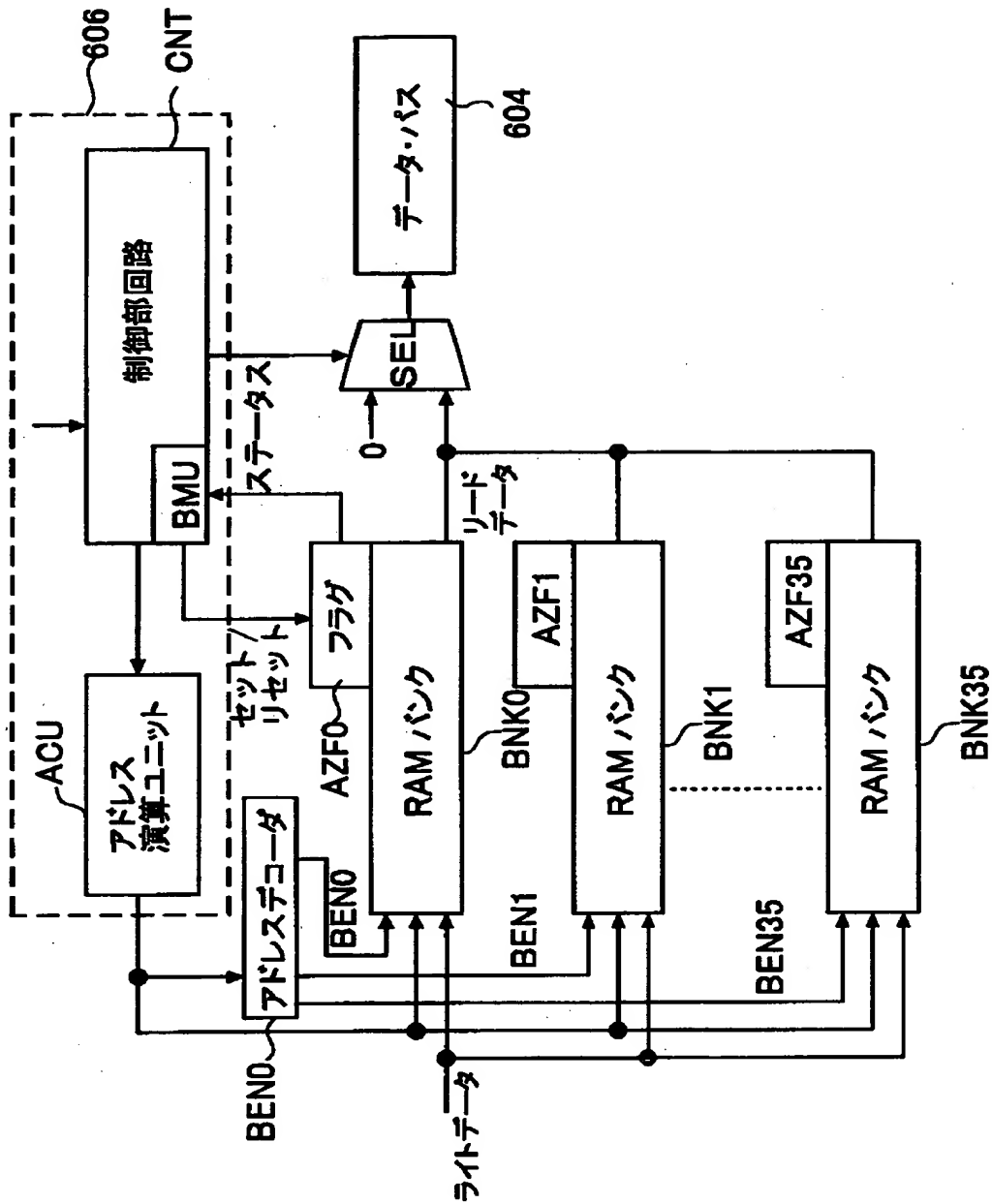
【図 7】



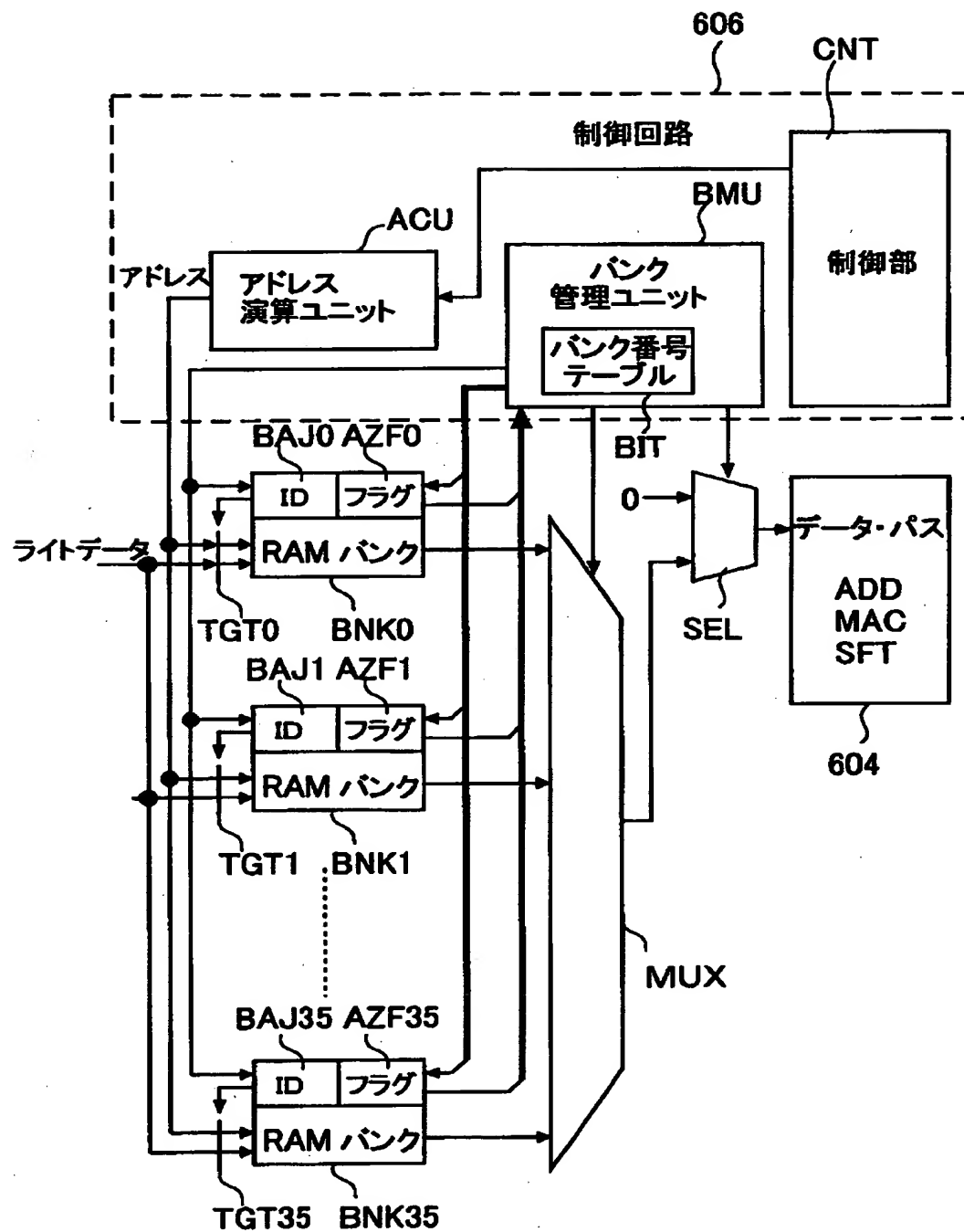
【図 8】



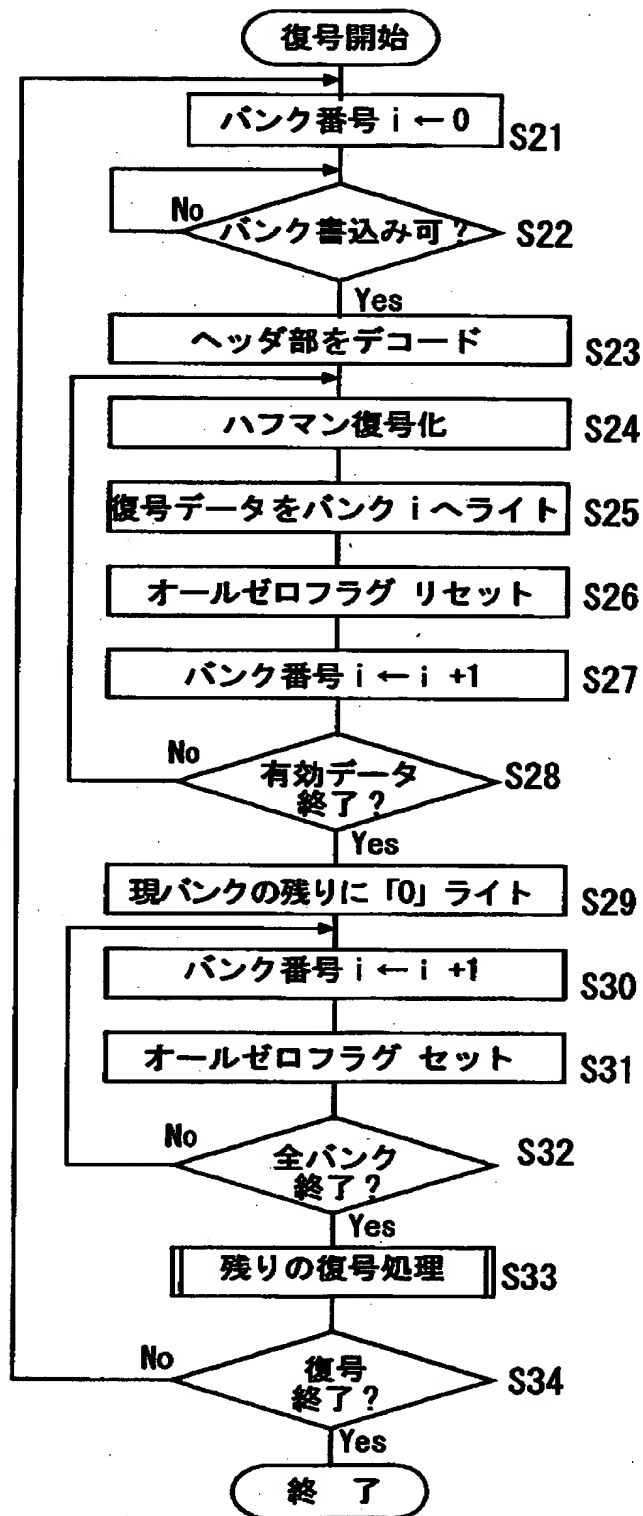
【図9】



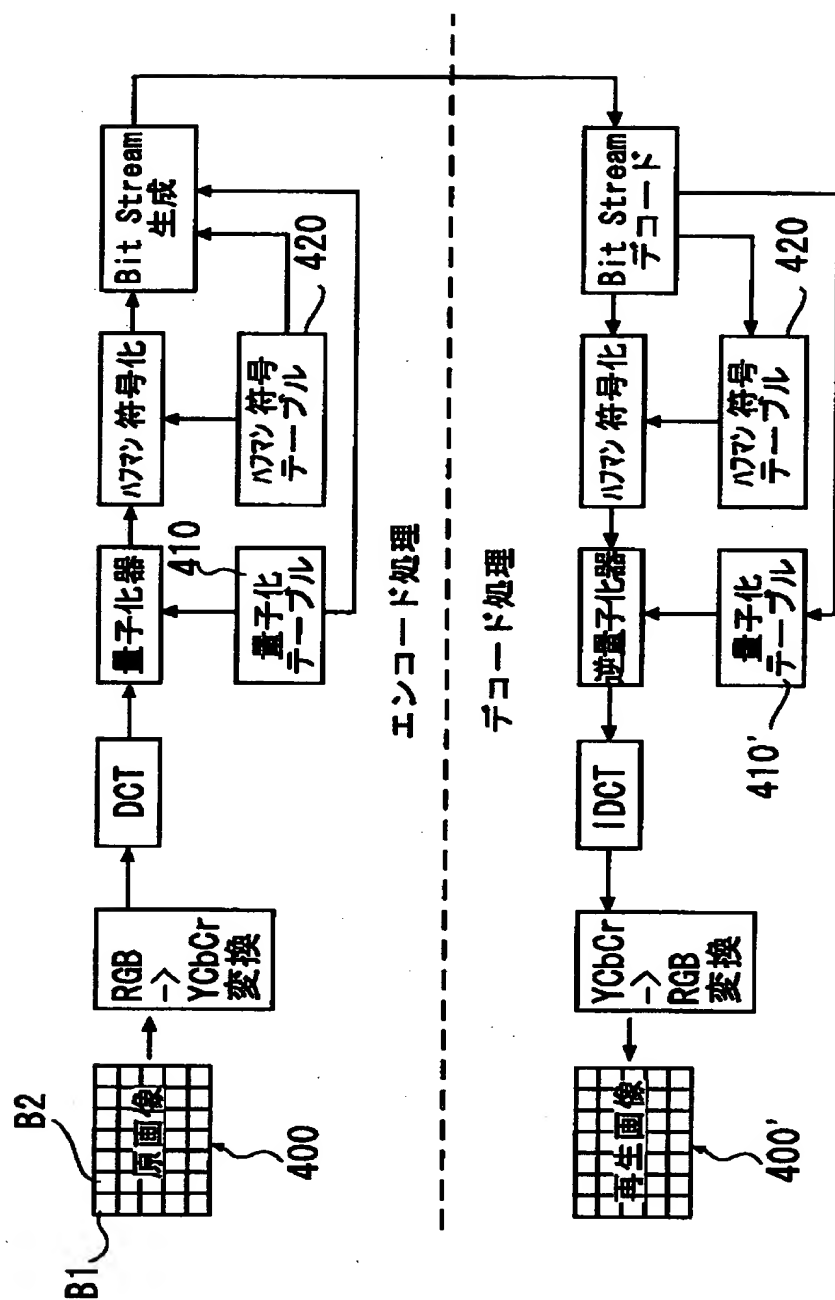
【図10】



【図 11】



【図 12】

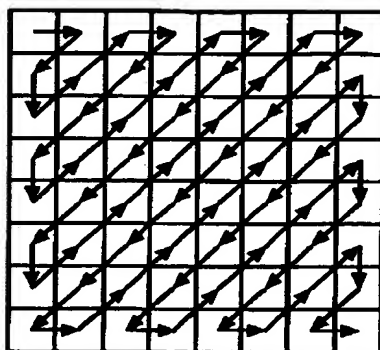


【図 1 3】

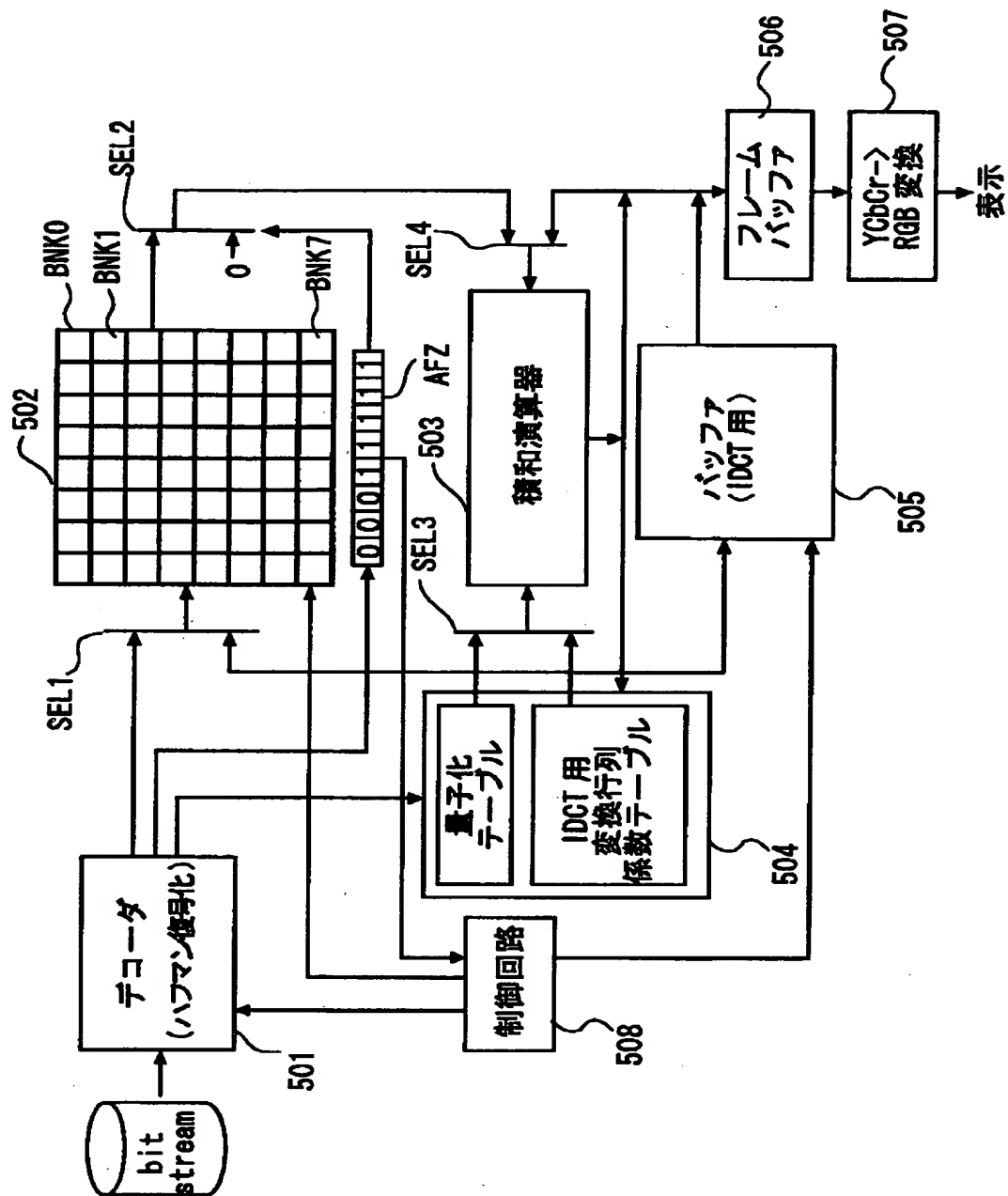
(A)

28	16	5	2	1	1	0	0
5	-4	-2	-1	-1	0	0	0
-1	1	-2	0	0	-1	0	0
3	-1	1	0	0	0	0	0
-1	1	-1	0	0	0	1	0
1	0	0	-1	0	0	0	0
1	-1	0	0	0	0	0	0
-1	1	0	0	0	0	0	0

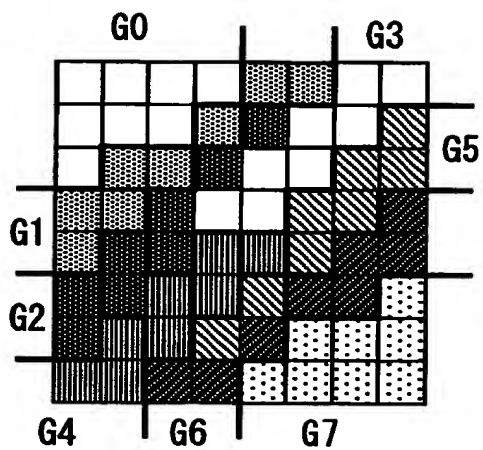
(B)



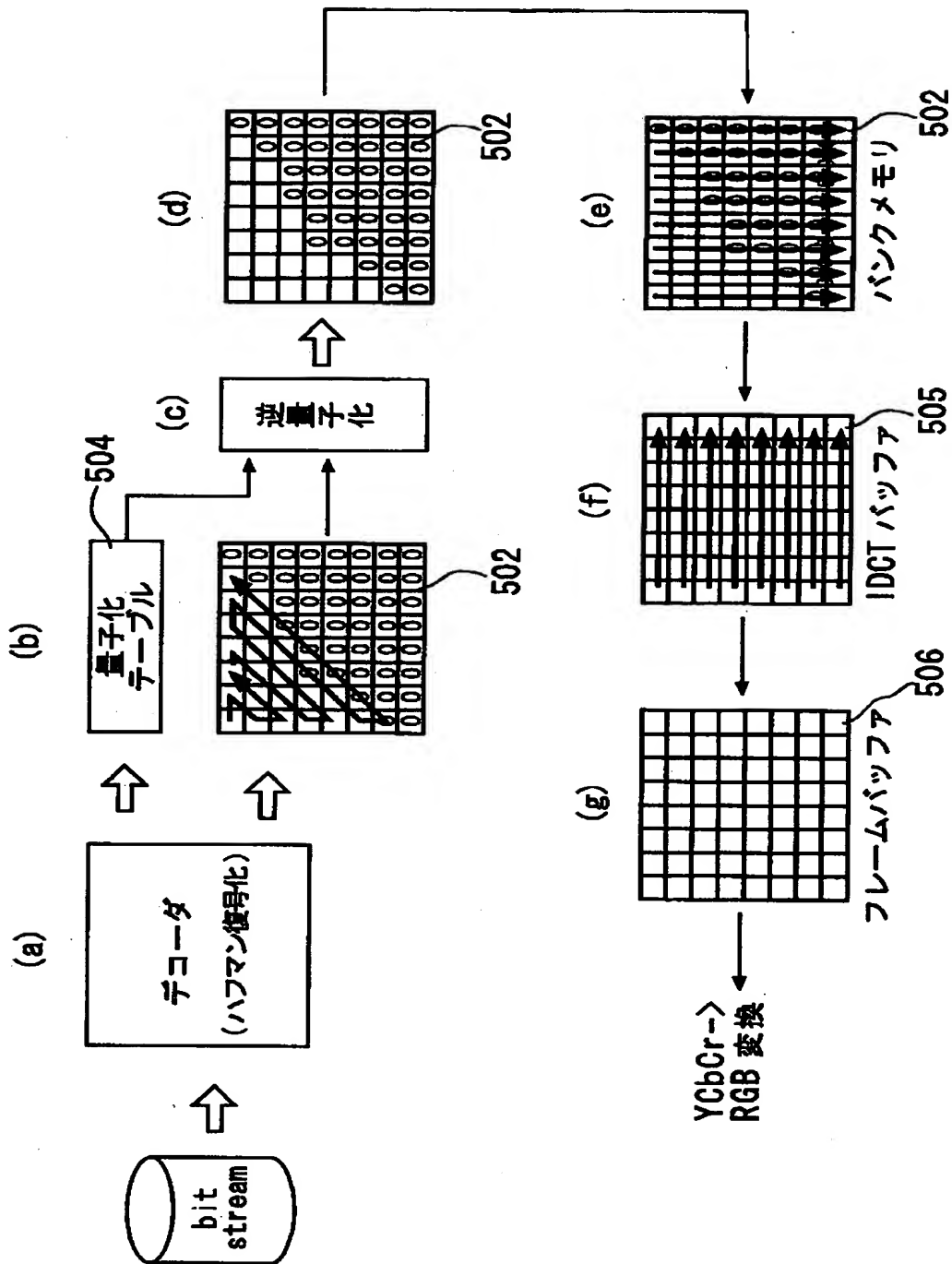
【図14】



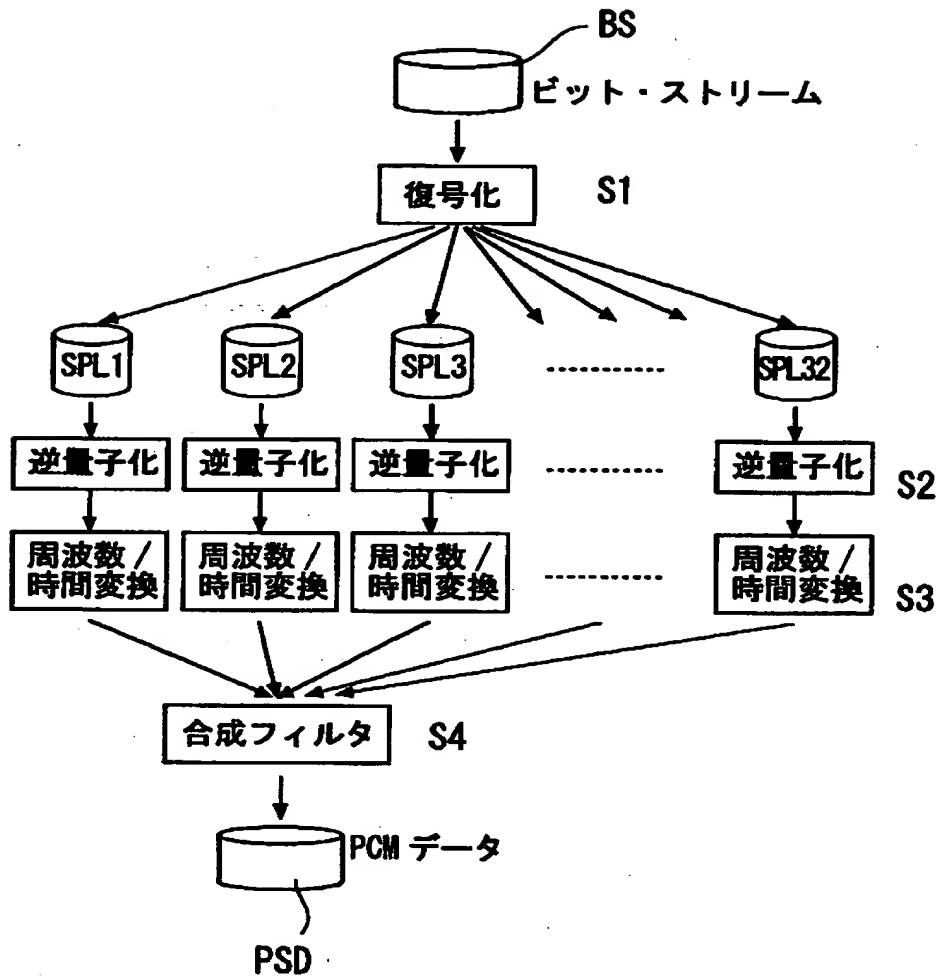
【図 15】



【図 16】



【図 17】



【書類名】 要約書

【要約】

【課題】 従来の圧縮されたオーディオデータの再生システムにおいては、伸張処理の際にすべてのビットが“0”である周波数帯のデータに関しても、RAMへのデータの書込み、読出し処理が行なわれていたため、RAMへのアクセスに伴う消費電力が必要以上に多くなっていた。

【解決手段】 MPEG規格あるいはJPEG規格に従って圧縮されたデータを伸張するシステムにおいて、圧縮されたデータを伸張する際に演算される値を格納するバッファメモリ（603）を複数のバンク（BNK）に分けて、各バンクごとにバンク内のデータがすべて“0”か否かを示すオールゼロフラグ（AZF）を設け、バンク内に書き込むべきデータがすべて“0”であるときはバッファメモリへの実際の手込みを行なわずにオールゼロフラグをセットするとともに、データ読出し時にはこのフラグを見てバンク内のデータがすべて“0”のときはバッファメモリからの読出しを省略するようにした。

【選択図】 図1

特2000-316093

認定・付加情報

特許出願の番号	特願2000-316093
受付番号	50001337974
書類名	特許願
担当官	第七担当上席 0096
作成日	平成12年10月18日

<認定情報・付加情報>

【提出日】	平成12年10月17日
-------	-------------

次頁無

出 願 人 履 歴 情 報

識別番号 [000005108]

1. 変更年月日	1990年 8月31日
[変更理由]	新規登録
住 所	東京都千代田区神田駿河台4丁目6番地
氏 名	株式会社日立製作所